IN THE CLAIMS

1. (Currently amended) A processor comprising:

controller circuitry operative to control ~~the~~ performance of a continuity check for each of a plurality of flows of protocol data units received by the processor; and

memory circuitry comprising a continuity check cache;

wherein the continuity check cache stores an identifier for each of a subset of the plurality of flows;

wherein the controller circuitry controls access to a set of continuity check counters comprising a counter for each of the plurality of flows;

the controller circuitry determining if a given flow for which a protocol data unit is received in the processor has a corresponding entry in the continuity check cache, and if the given flow has such an entry, preventing a corresponding one of the continuity check counters from being updated, and if the given flow does not have such an entry, clearing the corresponding one of the continuity check counters and storing a flow identifier for the given flow in the continuity check cache.

2. (Original) The processor of claim 1 wherein the memory circuitry comprises an internal memory of the processor, and the continuity check cache is implemented in its entirety within the internal memory.

3. (Original) The processor of claim 1 wherein the set of continuity check counters are stored in an external memory associated with the processor.

4. (Original) The processor of claim 1 wherein at least one of the continuity checks is performed in a manner compliant with an I.610 protocol.

5. (Original) The processor of claim 1 wherein at least one of the protocol data units comprises a cell.

6. (Currently amended) The processor of claim 1 wherein the continuity check cache has a capacity of M entries, ~~each~~ a given one of which may correspond to ~~a~~ the flow identifier, and the set of continuity check counters includes N continuity check counters, where M is substantially less than N.

7. (Original) The processor of claim 1 wherein one or more of the flows correspond to particular network connections.

8. (Original) The processor of claim 1 wherein each of the flows for which a flow identifier is stored in the continuity check cache has had its corresponding continuity check counter cleared upon receipt of a first protocol data unit for that flow within a specified time window.

9. (Original) The processor of claim 1 wherein each of the continuity check counters is configured so as to be incremented if one or more protocol data units are not received for the corresponding flow within a specified time window.

10. (Currently amended) The processor of claim 1 wherein in conjunction with [[a]] the continuity check performed for [[a]] the given flow the continuity check fails and a timeout indication is generated if the corresponding continuity check counter reaches a particular value.

11. (Currently amended) The processor of claim 1 wherein in conjunction with [[a]] the continuity check performed for [[a]] the given flow a corresponding one of the continuity check counters is reset only a single time for a plurality of protocol data units received by the processor for the given flow within a specified time window.

12. (Original) The processor of claim 1 wherein at least one of the continuity check counters comprises a multi-bit counter with each increment of the count representing a specified time window within a designated period of time for which the continuity check is performed.

13. (Original) The processor of claim 1 wherein at least one of the continuity check counters comprises a three-bit counter with each increment of the count corresponding to a time window having a duration of approximately 0.5 seconds.

14. (Original) The processor of claim 1 wherein the entries of the continuity check cache are cleared after expiration of each of a plurality of time windows for which the continuity check counters can be incremented.

15. (Currently amended) The processor of claim 1 wherein if the continuity check cache is full when one of the plurality of flows first arrives at the processor, a particular flow identifier from the cache is removed to make room for storage of [[a]] the flow identifier for the arriving flow.

16. (Original) The processor of claim 1 wherein the processor is configured to provide an interface for communication of the received protocol data units between a network and a switch fabric.

17. (Original) The processor of claim 1 wherein the processor comprises a network processor.

18. (Original) The processor of claim 1 wherein the processor is configured as an integrated circuit.

19. (Currently amended) A method for use in a processor comprising controller circuitry operative to control ~~the~~ performance of a continuity check for each of a plurality of flows of protocol data units received by the processor, the controller circuitry being further operative to control access to a set of continuity check counters comprising a counter for each of the plurality of flows, the method comprising the steps of:

storing an identifier for each of a subset of the plurality of flows in a continuity check cache; and

determining if a given flow for which a protocol data unit is received in the processor has a corresponding entry in the continuity check cache, and if the given flow has such an entry, preventing a corresponding one of the continuity check counters from being updated, and if the given flow does not have such an entry, clearing the corresponding one of the continuity check counters and storing a flow identifier for the given flow in the continuity check cache.

20. (Currently amended) An article of manufacture comprising a ~~machine-readable storage medium having program code stored thereon~~ computer readable medium having at least one computer program encoded therein for use in a processor comprising controller circuitry operative to control ~~the~~ performance of a continuity check for each of a plurality of flows of protocol data units received by the processor, the controller circuitry being further operative to control access to a set of continuity check counters comprising a counter for each of the plurality of flows, the at least one program ~~code~~ when executed in the processor implementing the steps of:

storing an identifier for each of a subset of the plurality of flows in a continuity check cache; and

determining if a given flow for which a protocol data unit is received in the processor has a corresponding entry in the continuity check cache, and if the given flow has such an entry, preventing a corresponding one of the continuity check counters from being updated, and if the given flow does not have such an entry, clearing the corresponding one of the continuity check counters and storing a flow identifier for the given flow in the continuity check cache.